IMPLEMENTING A QUANTITATIVE ANALYSIS DESIGN TOOL FOR FUTURE GENERATION INTERFACES

THESIS

Brandon M. Webster, 2Lt, USAF

AFIT/GSE/ENV/12-M08

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSE/ENV/12-M08

IMPLEMENTING A QUANTITATIVE ANALYSIS DESIGN TOOL FOR FUTURE GENERATION INTERFACES

THESIS

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Brandon M. Webster, BS

2Lt, USAF

March 2012

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GSE/ENV/12-M08

IMPLEMENTING A QUANTITATIVE ANALYSIS DESIGN TOOL FOR FUTURE GENERATION INTERFACES

Brandon M. Webster, BS
2Lt, USAF

Approved:

John M. Colombi, PhD (Chairman)	Date
Michael E. Miller, PhD (Member)	Date
Randall W. Gibb, Col, USAF (Member)	Date

Abstract

The implementation of Multi-Aircraft Control (MAC) for use with Remotely Piloted Aircraft (RPA) has resulted in the need of a platform to evaluate interface design. The Vigilant Spirit Control Station (VSCS), developed by the Air Force Research Laboratory, addresses this need by permitting the rapid prototyping of different interface concepts for future MAC-enabled systems. A human-computer interaction (HCI) Index, originally applied to multi-function displays was applied to the prototype Vigilant Spirit interface. A modified version of the HCI Index was successfully applied to perform a quantitative analysis of the baseline VSCS interface and two modified interface designs. The modified HCI Index incorporates the Hick-Hyman decision time, Fitts' Law time, and the physical actions calculated by the Keystroke-level model. The analysis indicates that the average time for the modified interfaces is statistically less than the average time of the original VSCS interface. These results revealed the effectiveness of the tool and demonstrated in the design of future generation interfaces or modifying existing interfaces.

To my wife

Acknowledgments

I would like to thank my thesis advisors, Dr. John Colombi, Dr. Michael Miller, and Col Randall Gibb, for their abundant support and incredible guidance throughout my thesis research. Without their knowledge, the depth of my research and level of my analysis wouldn't have even been possible. I credit each of them for challenging me to reach my fullest potential and increasing my level of education. I would also like to thank the 711th HPW and especially Dr. Michael Patzek for his hospitality. I am very grateful for the data supplied that became the focus of my research. This provision vastly improved the quality of my work and opened doors that wouldn't have been possible without his help. Lastly, I want to thank my amazing wife. Without her continuing support, I wouldn't be standing where I am at today. I am very grateful for all her love and patience throughout this rigorous program.

Brandon M. Webster

Table of Contents

Page

Abstract ... iv

Acknowledgments ... vi

Table of Contents ... vii

List of Figures ... ix

List of Tables ... x

List of Equations ... xi

I. Introduction ... 1

- General Issue ... 1
- Problem Statement ... 4
- Research Questions ... 5
- Research Focus ... 5
- Methodology ... 6
- Preview ... 7

II. Scholarly Article ... 8

- Introduction ... 9
- Background ... 10
  - *HCI Index* ... 11
  - *VSCS* ... 17
  - *MUSCIT* ... 18
- Method ... 19
  - *Input Data* ... 19
  - *State-based Graph* ... 19
  - *Affinity Matrix* ... 20
  - *Fitts' Law, KLM, and Functions* ... 21
  - *Modified HCI Index* ... 22
  - *Experimental Design* ... 23
- Results ... 24
  - *Statistical Test Results* ... 26
  - *Discussion* ... 28
- Conclusions ... 28

III. Conclusions and Recommendations ... 30

Chapter Overview 30
Conclusions of Research 30
Significance of Research 31
Recommendations for Future Research 31
Summary 32

Appendix A 33

Bibliography 43

List of Figures

Page

Figure 1: F-15 MFD Layout (US Gov't figure) 11

Figure 2: Vigilant Spirit Control Station Initial Startup 18

Figure 3: Graph of Baseline VSCS' Nodes and Edges... 20

Figure 4: Affinity Matrix Plot.. 21

Figure 5: State-based Approach... 25

Figure 6: Average Control Time with Noise 26

Figure 7: Regression Plot... 27

List of Tables

Page

Table 1: KLM Operators and Times 15

Table 2: Noise Floor 24

Table 3: ANOVA Configuration 27

Table 4: Tukey HSD Comparison 28

List of Equations

Page

Equation 1 13

Equation 2 13

Equation 3 14

Equation 4 14

Equation 5 22

Equation 6 23

Equation 7 23

IMPLEMENTING A QUANTITATIVE ANALYSIS DESIGN TOOL FOR FUTURE GENERATION INTERFACES

I. Introduction

General Issue

With the success of Remotely Piloted Aircraft (RPA) in the battlefield, the Department of Defense continually seeks to maximize the utility of this unique system in the Joint fight. The RPA is of the most solicited capabilities that the United States Air Force exploits to the Joint Force (USAF, 2009). As a result, the increasing demands for RPAs in unique military operations are exponentially growing. The RPA brings a multitude of roles to the warfighter including persistence, undetected penetration/operation, operation in dangerous environments (without putting a human in harm's way), and integrated "find, fix, finish" sensor and shooter capabilities on one platform (USAF, 2009). With the high demand of these systems in the battlefield, there is an urgency for technology exploration to fully utilize the current human-computer interface capabilities.

Exploring new concepts for the RPA system requires the Air Force to invest heavily in this type of research. The *United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047* addresses future plans for the RPA system that coincides with the Air Force vision. Being such a complex system, the RPA requires a host of highly-skilled individuals to operate each component from the Ground Control Station (GCS) interface to the individual sensors. Multi-Aircraft Control (MAC) is a concept discussed in the flight plan where one pilot controls multiple aircraft from a single ground station while maintaining situational awareness of the surroundings from each area of responsibility

(AOR). Situational awareness is a term originally coined in the aircraft pilot community which describes “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future” (Endsley, 1988). Development of such a capability would reduce the manpower required to support a given sortie rate or increase the sortie rate beyond those established by current manpower constraints. For the pilot to be effective in such a scenario, the pilot would require “automation with a clear and effective user interface” (USAF, 2009).

The design of MAC or any other future concept for the RPA system has to account for Human Systems Integration (HSI). Defined by International Council on Systems Engineering (INCOSE), HSI is the “interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice” (2007). The Air Force recognizes nine domains of HSI which include manpower, personnel, training, human factors, environment, safety, occupational health, survivability, and habitability (DoD, 2012). Human factors addresses the design of systems to improve the performance of the user within the systems (Hardman, 2009).The human factors domain is often broken down even further into categories including cognitive, physical sensory, and team dynamics (Hardman). Today, as the applications for computers have exploded in the recent decades, significant study has been performed in human-computer interaction (HCI) (Hardman). HCI typically refers to the design and optimization of user interfaces (UIs). In this same respect, as described by *United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047*, the ultimate success of any UAS will fully depend on the success

of the human interfaces. The interface designs should tightly integrate human considerations, including human limitations and capabilities, into the interface development.

The 711th Human Performance Wing (HPW) developed a software-based interface that conducts various HCI studies called the Vigilant Spirit Control Station (VSCS). VSCS is a research platform that can be used to assess and evaluate various human system interface concepts (Rowe et al., 2009). This testbed is used to simulate missions that include common RPA tasks, providing the ability to iteratively design and test future human interface concepts. Current studies associated with VSCS include Multi-UAV Supervisory Control Station (MUSCIT) (Patzek et al., 2008). Other studies conducted by the 711th HPW and studied using VSCS include the Cooperative Operations in Urban Terrain (COUNTER) (Feitshans et al., 2008). MUSCIT is focused on developing a display that allows a single operator to supervise multiple RPAs in a static, dynamic, and close air support mission. (Patzek et al.). The COUNTER program refers to layer sensing small Unmanned Air System (UAS) in urban terrain that are able to release Micro UAS's to generate closer displays at a lower altitude (Feitshans et al.). With VSCS' flexible software architecture, it is able to handle various environments and handle multiple programs for control of multiple vehicles of all types (Rowe et al.).

With increasing amounts of automation in the unmanned vehicle community, the fact is that "the operations of the vehicles always include a human component, and thus the need for a ground control station (GCS)" (Rowe et al., 2009). VSCS is designed to permit operator-vehicle interface technologies for managing, controlling and operating multiple RPAs with minimal crew size. VSCS can simulate the various missions with

multiple vehicles to permit pilots to interact with and provide feedback on the system to determine if certain capabilities such as MAC are manageable. This simulation allows for studies of new technologies and draws conclusions from experiments of experienced RPA pilots. Utilizing these conclusions, a more robust and efficient interface can be implemented.

Problem Statement

Currently VSCS offers no quantitative way to predict the pilot performance or length of time it takes RPA pilots to complete individual tasks. Instead, interface designers must rely on heuristic design principles to propose an interface, evaluate the interface through usability testing and then apply the lessons from these tests to propose further improvements to the interface. This evaluation of the interface is necessary since most UI design principles are ad hoc and based on experts' best guesses, rather than true data (Mayhew, 1992). Therefore, the usability and utility of an interface is not assured without an independent evaluation with representative users.

As a result of the need to evaluate each interface with existing tools, each iteration of the interface can only be evaluated through time-consuming and costly usability studies conducted with pools of representative pilots. This limitation suppresses the speed at which iterative solutions can be explored and lengthens the time necessary to field a more optimal user interface. Consequently, there is a need for a tool to evaluate a user interface that can predict pilot performance, pilot workload and the length of time required to complete a task. This tool would allow more rapid user interface iteration between usability tests. Such tools need to account for many domain-specific

considerations, including those requiring more rapid assessment of time critical or and are performed so infrequently that less efficient implementations of these tasks would hinder operator performance. Thus a modeling tool approach is needed to help quantify an operator’s performance under various manipulations of system interface designs.

Research Questions

The objective of this thesis is to identify quantitative methods for evaluating early interface designs or design modifications, such as those that might be applied within the VSCS. More specifically, the goal was to determine the average task times and an overall weighted average control time for a VSCS interface. To address this question, this thesis applies and extends the Human-Computer Interaction (HCI) Index (Hardman, 2009) to evaluate an existing VSCS interface and to demonstrate this method to evaluate alternatives to this interface. The research questions that were addressed include:

1. How can the HCI Index be applied to evaluate context-aware average control time of the interface?
2. What are modified interface designs that could reduce this average control time and potentially improve human workload?

Research Focus

The focus of this study revolves around the AFRL Vigilant Spirit Control Station. This particular user interface can be extrapolated and compared to more recent interface designs, but only Vigilant Spirit will be studied.

The input data leverages recent investigations of the AFRL Multi-UAV Supervisory Control Interface Technology (MUSCIT) program. This program is based on

“human systems integration; developing and integrating controls, displays, and decision support aids that enable a single operator control station to control multiple unmanned aerial vehicles” (Patzek et al., 2008). When attempting to assess multi-UAV control, “the development of a realistic and robust simulated operational environment” was utilized (Patzek et al.). To employ MUSCIT, experienced RPA pilots are placed in the VSCS simulation and their activities and performance are recorded via usability testing software. This data including mouse-clicks, markers, mouse location, time, and voice can be extracted for additional studies. For this particular study, the first segment of a simulated mission was quantified which included eight different pilots with the same mission but different map layouts. These eight pilots all had the same task of performing static Intelligence, Surveillance and Reconnaissance (ISR). During this part of the mission, ‘UAVs are often assigned to observe, monitor and/or track ground entities operating in a particular area of interest.” (Patzek et al.). Prior to ISR, each pilot had to setup the interface to their liking which left them with many options. This concept is becoming more dominant in user interfaces where each individual can customize their own interface, while being able to perform the same functions as others. This is different than traditional interfaces, which provide a common interface arrangement for every user.

Methodology

Beginning stages of this research began with recognizing the root problems of MAC through a discrete-event workload model of the MQ-1 Predator. Previous research revealed that high workload spikes during MAC involved the volumes of communication events (Schneider et al., 2011). After interviews and observations with experienced RPA

pilots in Creech AFB, the data revealed a desperate need for a GCS redesign before MAC could be effectively implemented. Therefore, the research was focused to investigate evaluation tools for future interfaces. This research selected the HCI Index to the VSCS for determining layout effectiveness (Hardman, 2009). Due to the differences of the VSCS interface to previous HCI Index applications, an updated approach was developed using state-based nodes to appropriately graph the VSCS interface. Recent research (Seibert et al., 2010) which added Fitts' Law and Keystroke-Level Model features to the HCI Index was also incorporated. This created a robust measure that estimates the average control time of the user interface. After having this baseline measure, two modified user interfaces were assessed to determine if interface options could be readily identified that could minimize average control times.

Preview

This thesis follows the scholarly article format that includes two separate research paper that stemmed from a study of MAC. (Schneider et al., 2011). Appendix A was accepted by the *Conference on Systems Engineering* and will be presented at the March 1012 conference in St Louis, MO. This paper investigates shifting communication between modalities in a MAC-enabled environment and in an attempt to mitigate the workload induced from communication events. This study first raised the question of MAC in VSCS since it is designed for multi-vehicle platforms. In an attempt to create a workload model for VSCS, task times had to be determined and separate research was applied. The subsequent Chapter II contains this work and has been formatted for submission to the *International Journal of Human-Computer Studies*.

II. Scholarly Article

For Submission to the International Journal of Human-Computer Studies

Quantitative Analysis of Human-Computer Interfaces (HCI): The Human-Computer Interface Design Tool

Brandon Webster, John Colombi, Michael Miller, Randall Gibb

Abstract

The graphical User Interface (GUI) revolutionized computing and has become our primary means of interfacing with computers. Although the GUI has been in existence for more than three decades, it has continued to evolve and the recent advent of low cost, large area LCDs enable interaction with GUIs on much larger areas than ever before possible. This large area enabled interaction techniques that were heretofore untenable. Unfortunately, there is a lack of early quantitative analysis to evaluate options within these a user interfaces. This paper explores the extension and application of the HCI Index, a human-computer interaction (HCI) tool that was originally used to measure menu-based multifunction in an aircraft cockpit. The HCI Index was modified to include state information that allows the tool to be successfully applied to a modern interface. Utilizing this new HCI Index, proposed interface designs were evaluated to estimate the average control time. This research measures the sensitivity of the HCI Index to the selected variations in the context-aware GUI behavior.

Introduction

Although originally applied on relatively small, monochrome displays, the power of the GUI has rapidly increased as the visual displays we use to view them improve at an accelerating rate. As early versions of the GUI were applied on small area displays, which restricted the number of items that could be represented to a user at any one time, early versions of the GUI often displayed relatively few items to a user and required the user to navigate through several pages of menus to access a large number of features. This paradigm has been recently challenged as rapid evolution in flat panel display technology has enabled affordable large area displays, capable of simultaneously displaying a large number of items to the user at any one time.

Increasing the display area through the use of larger or multiple monitors change the way users interact with an interface. With multiple displays, users tend to arrange windows within each display instead of across the boundaries of all monitors (Ashdown et al., 2005). Another aspect of current-generation interfaces is their ability to have flexibility inside windows. The ability to minimize, maximize, hide, and change the size of the interface layout allows several options for each user to complete any task. As well as flexibility, using the entire screen creates the opportunity to display an abundant number of functions and actions at once. More items are visibly displayed which potentially makes functions easier to find.

The goal of the present research was to apply Hardman's HCI Index with Ward's improvements to a newer-generation interface and to demonstrate its utility within this domain. With growing variables and tradeoffs, a tool for producing quantifiable results could serve to improve the rate of user interface evaluation. The Human-Computer

Interaction (HCI) Index is a metric that performs a "quantitative evaluation of layout effectiveness" (Hardman, 2009). Using the HCI Index, Fitts' time, and the Keystroke-Level Model (KLM) can permit the average control time to be estimated for a user layout (Seibert, 2010).

Applying the HCI Index to a future generation interface faced a set of challenges. These challenges arose from the abundance of options and flexibility of the VSCS interface. This paper discusses a process of how for improving human-computer interaction (HCI) between man and machine regarding a newly innovated interface.

Background

Prior researchers have attempted to determine the average control time for an interface. Hardman proposed the HCI Index, which incorporates the Hick-Hyman Law to determine the overall effectiveness of an interface layout (Hardman, 2009). Hardman's research focused on aircraft multi-function displays. Figure 1 shown below reveals the multi-function display that was used in Hardman's study (Hardman, 2009a). He also developed a hybrid algorithm that uses the HCI Index to predict an "optimal" layout. Unfortunately, Hardman did not build and test the new interfaces to prove the layouts were optimal or improved over the existing interfaces. Ward expanded upon Hardman's research and incorporated the average Fitts' Law time and the Keystroke-Level Model to Hardman's research, as well as applied the revised model to analysis of a graphical user interface (Seibert, 2010). Unfortunately, Ward also did not have the user input to confirm estimations of average control time she computed through this revised model.

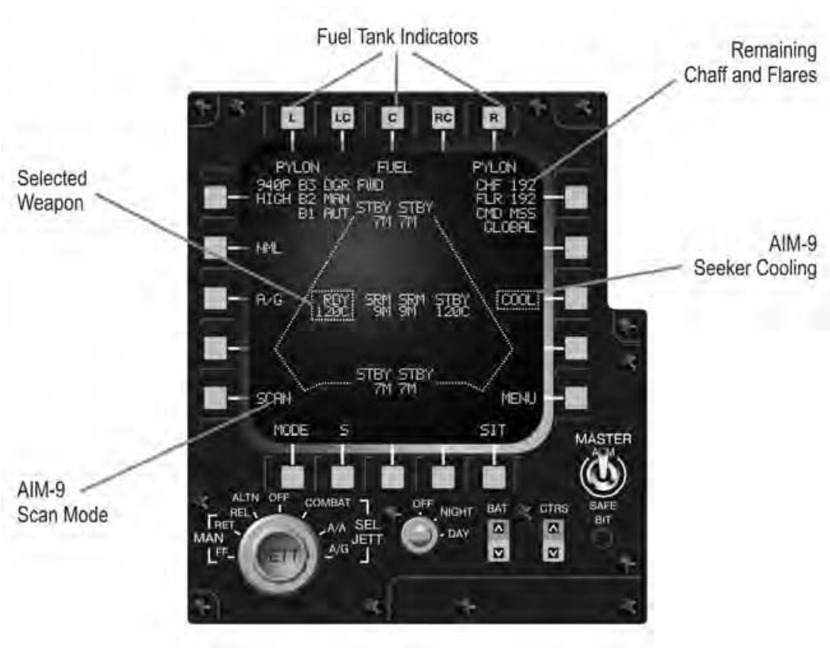

Figure 1: F-15 MFD Layout (US Gov't figure)

HCI Index

The HCI Index estimates the average time necessary to access a function in an interface where the each choice selection is assumed to be independent of past actions. Mathematically, this is using the theory of Markov chain where it is modeled with a graph consisting of nodes and connecting lines called edges (Hardman, 2009). To gain a clear perspective, an understanding of nodes, edges, and transitions between each must be grasped.

Nodes are the data displays outputs which can be referred to as a *page*. The output can be a combination of menu, options, functions, and information where each is modeled as a separate node from the layout (Hardman, 2009). Looking deeper, the menu options represent available transitions to separate pages as well as executed functions where the interface displays the same information. These are all separately modeled as different nodes in the graph. Edges are the interface inputs that are selected by the user

whether from menus or buttons, selectable icons, and voice recognition commands (Hardman). Simply put, edges are the transition of one node to a different node.

The graph of an interface layout consists of the nodes and edges defined previously. The representation of the graph is modeled by a directed graph (digraph). Self-loops indicate functions where an input (edge) creates the same *page* (node) transition (Hardman, 2009). Forming the graph can be done by creating an adjacency matrix of binary numbers (0 or 1). The adjacency matrix is best formed by listing every node and if an input (edge) exists, than the binary number is one. If an edge does not exist, the binary number is set to zero. As such, the adjacency matrix indicates the presence or absence of connections between different interface functions. The diagonal of the adjacency matrix is set to one if self-loop functions exist, but are set to zero if they do not.

Separate from the adjacency matrix exist the affinity matrix P. The affinity matrix represents the tasks as they relate to the adjacency matrix and can be formed by counting the number of times a representative group of users transitions between two nodes while using an interface. The affinity matrix is then set according to the elements corresponding to the nonzero elements of the affinity matrix to the appropriate counts from the adjacency matrix. Individual work flows can be counted from existing systems and the recommended method for nonexistent systems can be best formed from a task analysis (Stanton et al., 2005). The affinity matrix represents the joint probability of a transitioning to b, represented by $P(a,b)$. Once all transitions are counted, a weighted affinity matrix is created by normalizing the matrix by taking the sum of the matrix and dividing every $P_{i,j}$ by this sum. Doing this, the sum of the entire matrix equals 1.0.

The relation of the adjacency matrix and affinity matrix allows the evaluation of the HCI Index. Equation 1 shows the Hick-Hyman Law equation where *a* "describes the sum of those processing latencies that are unrelated to the reduction of uncertainty, such as execution or encoding time" and *b* equals "the amount of added processing time that results from each added bit of stimulus information to be processed" (Wickens et al., 2004).

$$RT = a + b \log_2 M \tag{1}$$

The Hick-Hyman Law states the uncertainty of stimulus events affects response time (RT), according to "the number of possible stimuli, the probability of a stimulus, and its context or sequential constraints" (Wickens et al., 2000).

Equation 2 shows the *weighted distance* from node v_0 to v_k, where both are considered two arbitrary nodes.

$$d_w(v_0, v_k) = \sum_{i=1}^{k} (t_i + (0.212 + (0.15)\log_2(d^+(v_{i-1}) + 1))) \tag{2}$$

where:

v_0, v_k = arbitrary nodes

t_i = the system processing delay associated with the edge on the minimum path

0.212 s = Simple reaction time (Phillips, 2000)

$d^+(v_{i-1})$ = the out-degree of the tail vertex of the i^{th} edge of minimum path

Using this information, the HCI Index now accounts for the adjacency matrix weighted by the affinity matrix shown in Equation 3 in milliseconds.

$$HCI = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} d_w(v_i, v_j) P_{i,j} \tag{3}$$

where:

N = the number of nodes

$P_{i,j}$ = affinity matrix

Since the foundation is now laid for the HCI Index, next was incorporating the physical key stroke times and pointing method. Fitts' Law states that the time T to acquire a target depends on its width W and the distance from the starting position to the target center (Fitts, 1954). Fitts' Law is relied on for predicting the time for pointing to an object with a given width and distance (Accot et al., 2003). Refining Fitts' Law to be applied to bivariate pointing, Accot and Zhai at IBM determined the best representation of Fitts' Law is presented by Hoffmann and Sheikh's data by the following equation:

$$T \approx -30 + 106 \log_2 \left(\sqrt{\left(\frac{D}{W}\right)^2 + 0.32\left(\frac{D}{H}\right)^2} + 1 \right) \tag{4}$$

where:

T = Fitts' Law time (in seconds)

D = distance of current pointing device to center of target

W = width of target

H = height of target

The KLM proposed by Card, Moran, and Newell (Card et al., 1983) measured the physical key-level actions in seconds. The key-level activities focused on for this particular study include placing hands to keyboard or mouse, a keystroke, typing a sequence of characters, pointing with pressing or releasing a mouse button, and clicking a

mouse button. Table 1 represents the time it takes to perform these actions. The original KLM study didn't contain a "mouse wheeling" operator, so the assumption was it took .1 seconds like the mouse button.

Table 1: KLM Operators and Times

Operators	**Description**	**Time (sec)**
K-Keystroke	Pressing button on keyboard	.28
T(n)- (n x K)	Typing sequence of *n* characters on keyboard	(*n* x .28)
B- Mouse Button	Press or release mouse button	.1
BB- Mouse Click	Click left or right mouse button	.2
H- Home	Home hands to keyboard or mouse	.4
*W- Mouse wheel	Wheel mouse forward or backward	.1

Tradeoffs

Within the realm of user interface design, there exists a tradeoff and compromise (Mayhew, 1992). Success of an interface depends on several areas including "functionality, performance, cost, reliability, maintenance, and usability" (Mayhew). From the user's perspective, often the faster any given task can be accomplished the better the interface. With current generation interfaces, bigger screens and the layout of information within them create significant tradeoffs. The three major variables that play a factor in response time within a graphical interface include the number of edges, Fitts' Law time, and the Hick-Hyman Law time.

Functional grouping can be commonly found in newer generation interfaces where common, multiple functions can be placed in a small area on the screen. The use of functional groups might reduce the number of items that a user might consider at once but may require multiple levels of decisions as the user must first select a functional group and then an item within the functional group.

The first variable, number of edges, is the simple reaction time of the user, which increases as the number of selections a user needs to make rises. Together these times can be summed to produce a total edge time (Hardman, 2009). For example, if a menu is displayed, the options that are presently visible describe edges. Increasing the depth of the layers would decrease the number of edges, due to the increasing menu hierarchy. Larger displays which present larger numbers of choices or edges permit an increase in the number of edges and an increase in functional grouping of edges would result in an increase in edges as well.

Fitts' Law time will vary greatly depending on the monitor size and number of monitors. If it is assumed that the size of buttons on a display are constant as the size of a display increases, then so will the average Fitts' time. Larger displays also provide the ability to display larger numbers of functions at one time which can reduce the number of selections a user needs to make to access a menu item. The increase of depth of layers and functional grouping has no bearing on Fitts' time.

The third variable is governed by the Hick-Hyman Law time which is the time required to choose an item from among a number of items. As the number of menu layers increase, the Hick-Hyman Law time will increase due to the amount of decisions that must be cognitively made. Larger screens also impact the Hick-Hyman time because the number of choices on the screen increases with the screen size. As the functional grouping increases, the Hick-Hyman time decreases due to the cognitive ability to make a decision more rapidly.

Knowing this information, increased screen size, which includes more items to select might indicates a higher level of Fitts' time and a higher Hick-Hyman response time for any given layer within an interface but because the interface is shallower, requiring navigation of fewer hierarchical menu layers it is unclear if the total edge time will increase or decrease with increases in display size. These tradeoffs change the overall use of an interface for a user and have to be consciously thought out when modifying or initially designing the interface. Table 2 provides a summary of tradeoffs when designing or modifying an existing interface.

VSCS

The basis of this study involves VSCS, an advanced graphical user interface (GUI) capable of supervising multiple vehicle platforms (Rowe et al., 2009). The overall purpose of this particular interface is to test new concepts that can potentially improve the human interaction with multiple vehicles. This interface is very flexible with the capability of supporting human centered experimentation. The human experimentation generally consists of experienced multi-vehicle pilots running through real world, simulated trials where the data is recorded for analysis. Using this data, it can be incorporated into upcoming concepts to enable a more advanced multi-vehicle supervisory control interface.

Figure 2: Vigilant Spirit Control Station Initial Startup

MUSCIT

An experiment consisting of experienced Remotely Piloted Aircraft (RPA) pilots was previously performed. The experiment consisted of a simulated, real-world mission that included providing static surveillance on two 24-inch display monitors. Prior to surveillance each pilot completed a set of general tasks before performing surveillance. These tasks were not limited to, but included maintaining possession of two unmanned vehicles, creating a boundary which the vehicles stayed within, selecting video sources, and the actual surveillance.

The trials for this study consisted of 8 different pilots that individually flew two Remotely Piloted Aircraft (RPA). The mission remained the same for every pilot, but the maps and locations changed, so not every mission looked exactly the same. The simulations of the pilots were recorded and the data was extracted to be analyzed for future use.

Method

Input Data

The input data used to create a graph was extracted from the simulated missions performed by the 8 experienced pilots. The data gathered from the recorded mouse-clicks led up to and included the first high-level task of the mission which was the static surveillance of a city.

State-based Graph

The number of interface options on the VSCS was quite large. Assuming the performance of the eight operators was a fair indication of the performance for the majority of the population, the graph was modeled from the task performance of the eight operators. This alleviated the need to model the entire VSCS, but gave the pieces necessary for this study. The *state-based graph* took into account the state-based approach which yielded a product that could be assessed using Hardman's HCI Index (Hardman, 2009). Figure 3 shows the *state-based graph* of VSCS with 38 nodes and 1,075 edges. This graphical representation doesn't take self-loop functions into account. Looking at this, it is easy to realize that modeling the whole VSCS to a graph would explode exponentially and would be infeasible to analyze.

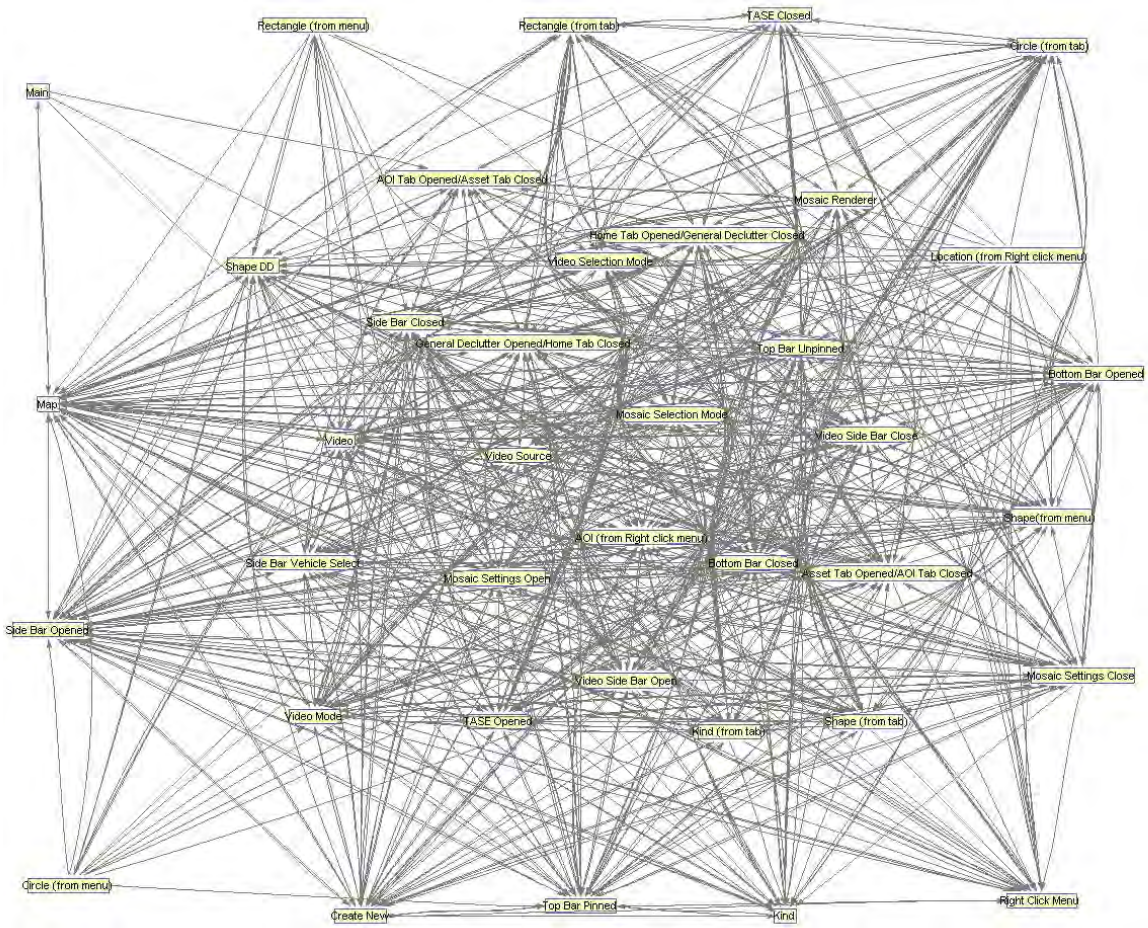

Figure 3: Graph of Baseline VSCS' Nodes and Edges

Affinity Matrix

After the *state-based graph* was completed, the *n* x *n* affinity matrix was formed where *n* is equal to the number of nodes in the graph. Using the input data, all transitions from one node to the next were counted and placed inside rho *P.* After the matrix contained all counts, rho *P* was weighted and the sum equaled to 1. The *Video* node contained the highest probability, 81%, which reveals that the pilots spend the majority of their time inside this node. Figure 4 truncates the *Video* spike, so other nodes can be viewed. As shown, no other single node has a probability greater than 3 percent.

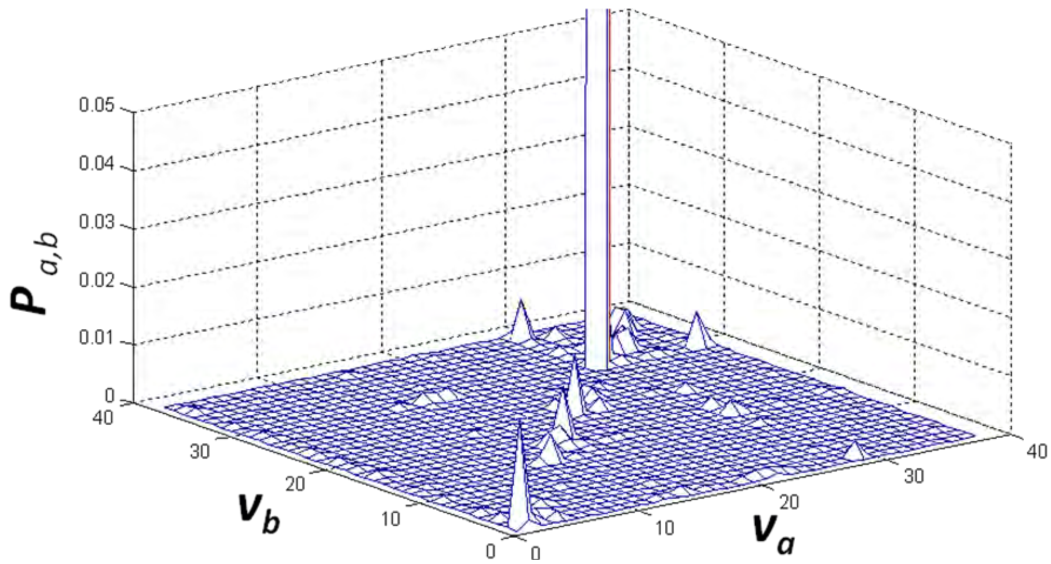

Figure 4: Affinity Matrix Plot

Fitts' Law, KLM, and Functions

To add Fitts' Law into the experiment, a ruler was used to conservatively measure one node to the next on the two 24-inch monitors. A worst-scenario was taken into account when measuring the distance between the node transitions. A worst-case width and height of the target was also used to determine the target size. Once this information was gathered, the Fitts' time was calculated with Equation 4 and placed into a *n* x *n* matrix, where *n* is the number of nodes in the *state-based graph*.

To incorporate KLM, a mouse-click (operator BB) was added to every node transition since the input data was all mouse-clicks. The self-loop functions required additional KLM estimates and Fitts' Law measurements. For example, the *state-based graph* contained an additional 30 functions that were represented as self-loops on the affinity matrix. To account for these functions, a worst-case scenario was used to measure Fitts' Law from the given node to the function. The KLM was then calculated

for the set of steps the pilot undertook to get through the function. Certain functions required typing, so this was accounted for as well.

To represent these self-loop functions, a n x n matrix was used, where n is the number of nodes in the *state-based* graph. The averages were taken for all functions along the diagonal of the adjacency matrix. For instance, *Video* is represented by the binary number 1 on the adjacency matrix diagonal which indicates there are self-loop functions. The total number of every self-loop function is averaged. The n x n matrix was then expanded with the averages.

Modified HCI Index

The design variable for this study was the *modified HCI Index* which is the average layout control time (in milliseconds) of VSCS. Using this as a measure, the goal is to have this value lowered in the overall experiment. To determine the average control time empirically, Fitts' Law time and the KLM had to be presented as an average from the entire layout as well as the HCI. Equation 5 presents the average Fitts' time weighted by rho ρ.

$$\overline{F} = \frac{1}{N^2}\sum_{i}^{n}\sum_{j}^{n}\left(TP_{ij}\right) \tag{5}$$

where:

T = Fitts' time
P= affinity matrix
N = number of nodes

Equation 6 shows the average KLM time weighted again by the affinity matrix.

$$\overline{K} = \frac{1}{N^2} \sum_{i}^{n} \sum_{j}^{n} \left(FP_{ij} \right) \tag{6}$$

where:

F = Function time matrix
P= affinity matrix
N = number of nodes

Equation 7 reflects the entire average control time for a current generation interface. This empirical evaluation estimates a good-fit measure that can quantify a layout. From here, the options are endless. Creating a layout baseline can easily be done and alternative layouts can be simulated either to create a new design or tweak the existing interface layout.

$$Modified\ HCI = HCI + \overline{F} + \overline{K} \tag{7}$$

Experimental Design

To determine if the interface could be improved, two separate layouts were proposed to test if a lower average control time could be produced. The first layout was changed by removing four nodes and combining them with others. It was observed that during the experiment, the pilot would always have to click to maximize the settings menu, so this graph represented the settings menu as always opened without the option of closing it. This layout had 34 nodes and 840 edges. Observations of the interface also indicated that the pilot had multiple options to complete a particular task, so the interface was assumed to be simplified to reduce the number of options and remove an additional 6 nodes. The second layout had 28 nodes and 646 edges. Both these layouts were then used to test against the original layout.

Since this is a model, there has to be an assumption that the input data is not 100% accurate. To understand the effect of this assumption, the input data can be modified to help improve its real world value. Different variations of random normal noise were added to the original counts in the *P* matrix. Table 2 shows the amounts of noise added in the experiment.

Table 2: Noise Floor

σ	.00025	.001	.001	.01	.01	.01	.01
μ	.00025	.001	.005	.01	.02	.03	.05

Incorporating all these factors, a series of 1,000 replications of the *modified HCI Index* was evaluated against each noise level. This ensured the results were statistically sound.

Results

Using Hardman's (2009) research, the implementation of Multi-functional display graph design couldn't be applied to a future generation interface. Incorporating the idea of nodes and edges in a digraph, a new approach of *state* was proposed. Figure 5 reveals the notion of *state*, where current generation interfaces have embedded menus that can "remember" information even if out of sight. This information can be hidden and restored, so there has to be a way to describe this instance of the system. This is done through the idea of a *state*.

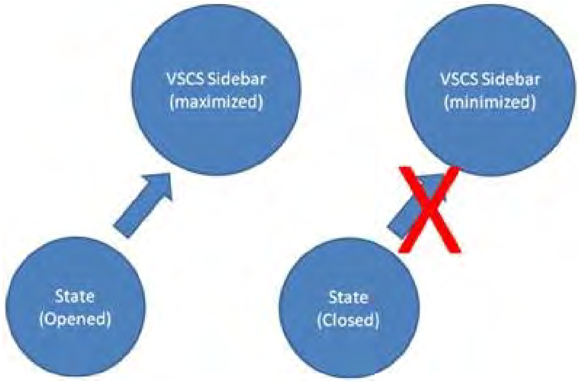

Figure 5: State-based Approach

A *state-based* approach has its limitations. There is no way to classify if a *state* is opened and remains opened in the affinity matrix. For instance, if there is a transition to an opened node and another transition returns to this opened node than the HCI Index is still calculated. There is an assumption that these returned, opened nodes still have a cognitive measure associated with them since VSCS has large displays, which present a number of options.

Using the idea of *states*, the *modified HCI Index* did estimate an average control time for the three layouts. Figure 6 displays the time with the noise floor on a logarithmic scale.

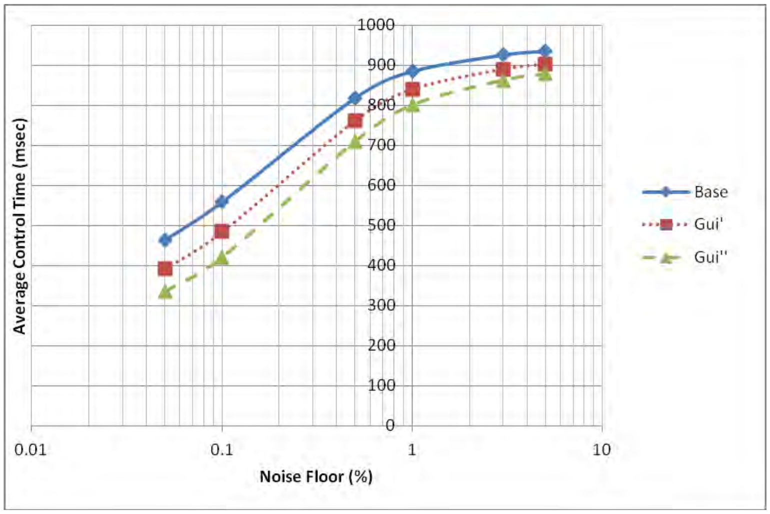

Figure 6: Average Control Time with Noise

The revealed general trends shown in Figure 6 indicate that as nodes were removed in the layouts, the *modified HCI Index* decreased. This would provide the pilot with a quicker average control time for performing the task in the study.

Statistical Test Results

To confirm the results, a two-way crossed Analysis of Variance (ANOVA) was performed. Since the effect of noise floor on average control time was clearly nonlinear, as shown in Figure 6, the noise floor conditions were logarithmically transformed as shown in Figure 7. As shown, this transform yielded a somewhat more linear function between noise floor and average control time such that the data meets the assumptions of the ANOVA. Table 3 shows the ANOVA configuration that was used to test the effects of the experiment.

Table 3: ANOVA Configuration

Response	Modified HCI Index
Factor	Gui Layouts
Factor	Logarithmic Noise Floor
Cross	Gui Layouts * Log Noise Floor

The ANOVA indicated that the GUI layout (F = 5596, p <=0), log Noise Floor (F = 418996, p<=0) and their interaction (F = 263, p <= 0.0001) were all significant. As shown in the regression plot of Figure 7 and confirmed with the ANOVA, noise floor had a large effect on average control time than did the GUI, with average control time generally increasing with the level of noise that was added. The residual plot indicates how they tend to shrink with a higher noise level.

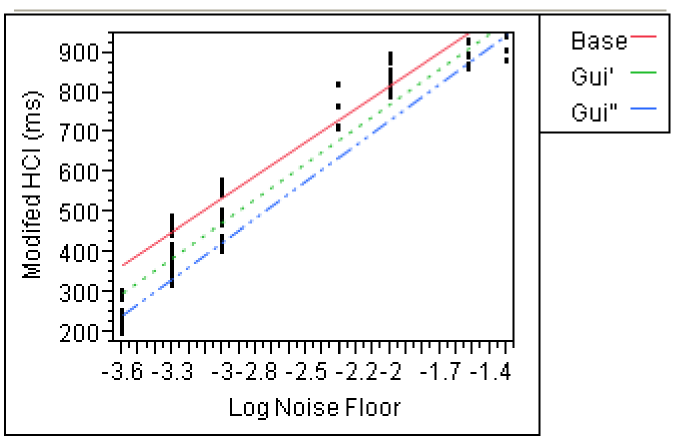

Figure 7: Regression Plot

Generally, the Base GUI required a longer control time than GUI', which had a longer time than GUI''. A Tukey HSD test is shown in Table 5 below to describe the difference in means adjusting for multiple comparisons. As expected, Base-Gui" had the

largest difference of 96.29 ms and Gui'-Gui" had the smallest difference of 43.67 ms. The p-values statistically shows there is a significant difference between these the 3 different levels.

Table 4: Tukey HSD Comparison

Level-Level	Difference	Lower CL	Upper CL	p-Value
Base-Gui"	96.29	94.16	98.43	0.00*
Base-Gui'	52.63	50.49	54.76	0.00*
Gui'-Gui"	43.67	41.52	45.80	0.00*

Discussion

Equation 7 revealed the mathematical formulation to calculate the average control time of a layout. Upon further analysis of these individual values, it was discovered that the HCI Index accounted for the majority of the empirical estimate. The original VSCS layout's HCI Index weighed in at 99.85%. The Fitts' time ($\overline{F}$) was at approximately 0.03% and the function time was at 0.12% ($\overline{K}$). This was true for the implemented noise cases as well. The average Fitts' time was lower than expected, but this could be due to the fact that in the majority of the time, the pilots stayed in the same node. When weighted by the affinity matrix P¸ Fitts' time and KLM have small contributions due to zero's in the diagonal of the affinity matrix.

Conclusions

In this paper, we have introduced a *modified HCI Index* which has estimated the average layout control time for pilots utilizing simulated runs from experienced RPA

pilots on the VSCS. Before the *modified HCI Index* was able to be applied to a future generation interface, a *state-based* approach had to be implemented. This approach took the interfaces' embedded menus and inner windows that could minimize/maximize into consideration. This quantitative tool can be used on current generation interfaces prior to design or to modify the existing system. With each individual design technique, there will be tradeoffs that have to be taken into consideration before any change is incorporated. These particular tradeoffs such as edge time, Fitts' time, and the Hick-Hyman time will change dependent on the system. When trying to reduce the user's time with the current VSCS interface, the easiest approach was to remove available options and minimize the functions on the interface. As the results show, this improved the average control time and helped improve the interface overall for the user.

III. Conclusions and Recommendations

Chapter Overview

This chapter reviews the original research questions described in the introduction and relates them to the findings in Chapter 2. After discussing the original research questions, the significance of research is discussed. Lastly, recommendations for future research and the summary will be presented.

Conclusions of Research

The original research questions were to determine if the HCI Index could be applied to evaluate the average control time of the interface. The second question was to determine if different layouts exist that could potentially improve the overall average control time.

We were unable to apply Hardman's HCI Index as it was applied to Multi-Function Displays. A *state-based* approach to computing this metric was defined and applied to calculate the HCI Index value for the VSCS.

Different layouts were presented, which did improve the user's average control time. The first layout contained the original GUI of VSCS and presented no changes. There were tasks that pilots performed which contained common actions that they tended to follow. Using this knowledge, the second layout combined a group of these common nodes into a similar node. This resulted in a faster average control time compared to the first. Lastly, there were two ways for a pilot to perform a same function inside VSCS.

Removing one of the options for two paths, the third layout was created. The last layout was more improved than the second and the first in regards to average control time.

Significance of Research

The research presented in this study concentrates on ways to improve the VSCS. The lessons learned can be applied to VSCS and improve the average control time for the pilot. As discussed earlier, tradeoffs will occur for every change that is made to the interface and this has to be well thought prior to modifications.

Advancing past VSCS, this research could be applied to any future generation interface in the Air Force or DoD. In the initial stages of development, a GUI can be well thought out and tested prior to fielding. This tool is not limited to only the VSCS, but expands past this interface.

Recommendations for Future Research

There are several options for future research in the particular area. The original idea was to have a flexible discrete-event model on VSCS using empirically estimated task times. The *modified HCI Index* currently predicts the average control layout time instead of task times individually. Now that the HCI Index can be applied to a future generation interface, the next step should be to have these task times calculated individually with specific Fitts' and KLM times. After an estimated task time exists, use these for a discrete-event simulation. This would provide another level of quantitative analysis which could be useful for evaluating an interface from a different perspective. Another useful study would be to investigate Fitts' Law time on larger displays. The typical user lifts their hands when moving mouse over large area, so determining this

frequency would result in a better design tool. Lastly, validating the *modified HCI index* on VSCS would be useful in determining its validity. Correlating the MOE/MOPs with the average control time would be beneficial for tweaking the tool for statistically significant results.

Summary

The research presented in this thesis started by examining MAC limitations. The original discrete-event simulation study disclosed unpredictable communication spikes that had to be solved to reduce a pilot's workload. After conducting a study on the communication spikes, the GCS stood out as being an area of dire improvement. Using this knowledge, VSCS was discovered to be a solution for the GCS, so an attempt was made to build a discrete-event model around this interface. The model required data to predict individual task times, so the HCI Index would have been a perfect fit. Using Hardman's HCI Index, a new *state-based HCI Index* was presented that would undertake future generation interfaces. This paper unveils the process used to apply a *state-based* approach. Using Ward's method to incorporate Fitts' time and KLM times into the HCI Index, a *modified HCI Index* was introduced. This empirically estimates the average control time of a layout. This time can decrease the time a user has to perform tasks and gives an overall estimate for new layouts proposed.

Appendix A

Submitted to Conference on Systems Engineering Research (CSER) 2011

Allocation of Communications to Reduce Mental Workload

Travis Pond, Brandon Webster, John Machuca,

John Colombi, Michael Miller, Randall Gibb

Abstract

As the United States Department of Defense continues to increase the number of Remotely Piloted Aircraft (RPA) operations overseas, improved Human Systems Integration becomes increasingly important. Manpower limitations have motivated the investigation of Multiple Aircraft Control (MAC) configurations where a single pilot controls multiple RPAs simultaneously. Previous research has indicated that frequent, unpredictable, and oftentimes overwhelming, volumes of communication events can produce unmanageable levels of system induced workload for MAC pilots. Existing human-computer interface design includes both visual information with typed responses, which conflict with numerous other visual tasks the pilot performs, and auditory information that is provided through multiple audio devices with speech response. This paper extends previous discrete event workload models of pilot activities flying multiple aircraft. Specifically, we examine statically reallocating communication modality with the goal to reduce and minimize the overall pilot cognitive workload. The analysis investigates the impact of various communication reallocations on predicted pilot workload, measured by the percent of time workload is over a saturation threshold.

Introduction

Over the past several decades, the US Air Force has harnessed and exploited the immense tactical power that middle and high-altitude Remotely Piloted Aircraft (RPAs) bring to the battlefield. As a consequence, the demand for RPA operational support continues to increase. It is important to realize that RPAs are part of a complex system. The system has many components including one or more air vehicles, ground control stations (GCS) for both primary mission control and takeoff/landing, a suite of communications (including intercom, chat, radios, phones, a satellite link, etc), support equipment, and operations and maintenance crews [1]. Assets and requisite resources to support those operations are limited and personnel resources, particularly RPA pilots, often prove a nontrivial constraint. This inevitably leads innovators to seek out RPA force-multiplying efficiencies to assist in bridging the resource/demand gap. One such efficiency being pursued is simultaneous control of multiple aircraft by a single pilot, or Multi Aircraft Control (MAC). This concept of operations has been documented in the US Air Force UAV flight Plan [2].which calls for future systems in which a single pilot will simultaneously control multiple RPAs to enable increased aerial surveillance without increasing pilot manpower requirements. Previous research on the cognitive workload experienced by pilots during MAC indicated that frequent, unpredictable, and oftentimes overwhelming volumes of communication events can produce unmanageable levels of system induced workload for MAC pilots [3]. To further investigate this identified problem, our study makes use of IMPRINT Pro, a Multiple Resource Theory (MRT) based dynamic, stochastic simulation to analyze impacts to cognitive workload by a disciplined communication modality reallocation construct.

Background

In the RPA domain, communication is a continuous and demanding process. Crews must track, at a minimum, information regarding weather, threats, mission tasking, mission coordination, target coordination, airspace coordination, fleet management, and status and location of any friendly units. The RPA pilot is not only responsible for aircraft control but is also a critical member in a multi-path communications infrastructure [4]. In the ground station, communication with the pilot takes place in one of two modalities: textual chat window(s) or the speech-based radio systems. At any given moment, a pilot may need to monitor multiple chat windows and listen to numerous parties operate over the radio. The multitude of communication sources and different media coupled with the quick inter-arrival rate of these events during a dynamic scenario drives an incredible cognitive workload for the pilot.

Cognitive or mental workload expresses the task demands placed on an operator [5]. Calculation of task demand, or task load, often considers the goals of the operator, the time available to perform the tasks necessary to accomplish the goals, and the performance level of the operator [6]. Therefore, workload increases when the number or difficulty of tasks necessary to perform a goal increase, or when the times allotted to complete these tasks decrease. Assuming that the operator has a limited amount of mental resources (e.g., attention, memory, etc.) that he or she can utilize to complete the necessary tasks, mental workload corresponds to the proportion of the operator's mental resources demanded by a task or set of tasks. Several methods have been employed to measure and quantify mental workload over the past four decades and have been summarized in numerous publications [5,7,8]. The current analysis incorporates Multiple

Resource Theory (MRT) into the workload calculations to account for channel conflict driven workload.

As a theory, MRT purports the existence of four mental dimensions (or channels) available to process information and perform tasks. The dimensions include processing stages, processing codes, perceptual modalities and visual channels. These channels are allocated to concurrent tasks with the difficulty of the tasks and the demand conflict between channels driving the overall mental workload value [9]. MRT accurately describes the concurrent nature of tasks imposed on an RPA pilot (performing primary tasks while communicating and monitoring communication) and is therefore an appropriate theory to apply to the present analysis.

Method

Therefore, the specific channels employed by the modeled communication events are highly relevant to the MRT workload calculations. As communication events begin to conflict with existing work activities on the various channels, the calculated overall cognitive workload will account for such conflicts. This construct enables the analysis to address the question of whether or not adjusting the intentional allocation of communication events to particular modalities will be able to meaningfully affect overall cognitive workload.

Model

A previous model of pilot mental workload [3] was utilized to understand the impact of communications modality. This model employed functional analysis and task allocation to construct an executable architecture of the multiple RPA system. This

architecture was then replicated within the Improved Performance Research Integration Tool (IMPRINT) to estimate the pilot's workload under various mission segments, such as handover, transit, emergency, benign and dynamic surveillance, etc. This model relied on subject matter expert input to develop distributions for the length, frequency, and difficulty of the events that induce workload on the pilot. The original research on this model indicated that workload was particularly high during what were termed dynamic mission segments. These mission segments often involve high levels of communication between the pilot and external actors to facilitate the tracking or observation of moving targets. High levels of communication resulted in particularly "high" pilot workload while operating a single aircraft and, "excessive" workload while controlling multiple dynamic-mission aircraft. The original research indicated that a reduction in pilot workload imposed by communication would be necessary to facilitate MAC.

To understand the potential impact of communication modality on operator workload, the communications portion of the earlier workload model was modified to permit communications events to be reallocated to alternate communications modalities. The revised model permits communication events that were originally allocated to the auditory channels where the operator listens and speaks to the visual and fine motor channels where the operator reads and types, or vice versa.

Figure 1 depicts the high level structure of the revised communications model. The gray boxes indicate model elements that were added to facilitate this particular evaluation. Communication events are generated with a mission segment dependent frequency and their interarrival times are exponentially distributed. In the original model, as a communication event is generated, it is assigned as either an auditory event or a text-

based event with 25% of the events being allocated as auditory events and the remaining allocated as text events. Half of the auditory events then required the pilot to talk or listen while 90% of the text events required the pilot to read while only 10% of the events required the pilot to type a response.

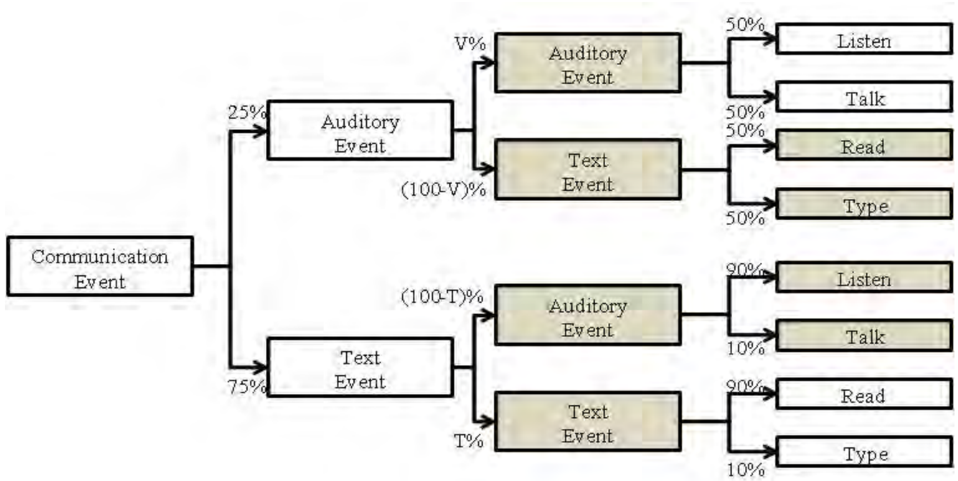

Figure 8: Modified communication model of pilot workload

To conduct the current evaluation, the model was modified as shown above. The auditory and text events shown in *gray* have the potential (through a notional device or software) to either pass an auditory or text event as a respective auditory or text event or to convert an auditory event to a text event or convert a text event to an auditory event. With this modification, it is assumed that the characteristics of the communication are due to communication needs, such that if a text event in the original model had a 90% chance of providing an input to the pilot and only a 10% chance of an output to the pilot, a text event converted to an auditory event has a 90% probability to require the pilot to listen and only a 10% probability to require the pilot to talk. The parameters *V* (for Voice reallocation) and *T* (for Text reallocation) provide the ability to convert auditory or text

events to its compliment. If *V* and *T* are both 100%, the revised model is the same as the original model. Reducing either of these parameters permits a portion of one type of communication event to be reallocated to the complimentary communication event. Although not shown, it is then assumed that some percentage of the final events generate a repeat communication event, indicative of a continued conversation. This aspect of the model was not changed.

Experimental Design

For this paper, a total of six "levels" of voice/text allocation were selected such that the percent of voice communication were varied between 0 and 100 percent. For levels of voice communications less than 25%, *V* was varied while *T* was maintained at 100%. However, for levels of voice communications greater than 25%, *V* was maintained at 100% while *T* was varied to achieve the desired communications levels. All analysis was performed for a 10 hour dynamic mission segment with a single pilot operating the aircraft. Although IMPRINT does not currently have built-in Monte Carlo functionality for the metrics of our concern, an external batch application was developed to automate replications. A total of 10 replications for each of six levels using 10 different random number seeds were performed to gather the output data.

The output of the IMPRINT model was analyzed to determine the proportion of time that the operator would experience workload values over a specified task saturation threshold. A workload value of 60 was calibrated to be about the 90% of operator "red-line", which indicates the workload value a pilot can experience without degraded performance [10]. The mean and variance across the 10 replications for each communication ratio was calculated. Analysis of Variance (ANOVA) and the Tukey

post-hoc tests were employed determine the statistical differences between the average of percent time over threshold.

Results

Figure 2 shows the percent time over threshold as a function of the percentage of voice communication. A one way ANOVA indicated a significant effect of the percent of voice communication upon the percentage of time over threshold ($p < 0.001$). As shown in Figure 1, the percent of time over threshold is reduced as the percent of voice communication is increased from 0% to 40%. At 40% voice communication the percent time over threshold is reduced to 24.5% compared to 33.1% with 0% voice communication. This change is statistically significant. The change in percent time over threshold is statistically insignificant as the percent of voice communication is increased from 40% to 60%. This trend indicates that pilot workload is reduced by the use of both auditory and text-based communications in this system.

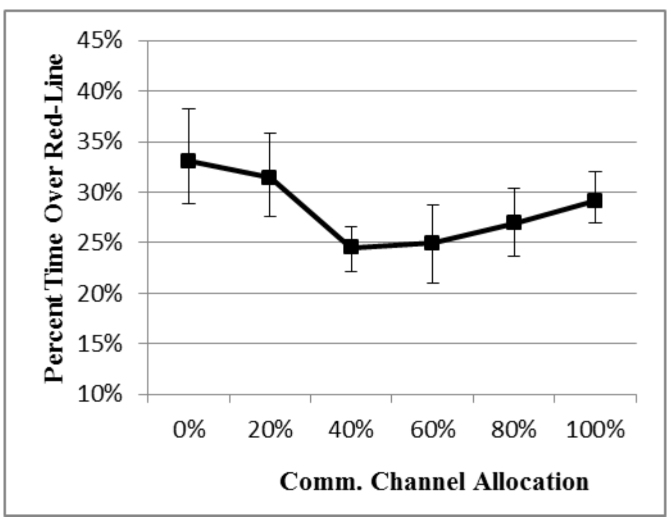

Figure 9: Percent Time Over Threshold as the percentage of reallocated voice events

Results further show that the percent time over threshold is greater at 0% voice than at 100% voice communications. This might have been expected as reading and typing likely conflicted directly with other tasks being performed by the pilot, including visually monitoring the status and manipulating the controls of the RPAs. As such workload is highest when all of the communication is allocated entirely to the visual channel.

Conclusions

The model indicates that by deliberately allocating communication between auditory and text-based modalities the pilot's workload and particularly the percentage of time the pilot operates beyond their task saturation red-line can be statistically reduced. The model shows that the percent of time over red-line is greatest when all of the communication is allocated to the text-based communications such that zero percent of the communication is allocated to voice. This type of communication is most likely to conflict with other tasks involving the visual system to monitor the RPA and the small motor system, which is used by the pilot to control the RPA. As communication events are moved from text to auditory, the workload decreases. However, as more communication is moved to the auditory channel, the percent of mission time over the red-line to increases. The increase likely occurs as the auditory tasks begin to overlap and conflict with one another to increase workload. There appears to be an optimal allocation of communications between voice and text modalities to achieve the lowest workload given a constant traffic load. Future research will examine dynamic reallocation of modalities.

References

[1] USAF MQ-1B Predator Fact Sheet. Air Combat Command, Public Affairs Office, 130 Andrews St., Suite 202; Langley AFB, VA 23665-1987. Sep 2010

[2] USAF, Headquarters. USAF Unmanned Aircraft Systems Flight Plan 2009-2047; Washington, DC; 2009.

[3] Schneider M, McGorgan J, Colombi J, Miller M, and D Long. Modeling Pilot Workload for Multi-Aircraft Control of an Unmanned Aircraft System. *Proceedings fo the INCOSE International Workship*; Denver, CO; 2011.

[4] MITRE. (2009). *Air force unmanned aircraft systems unconstrained architectures,* USAF.

[5] Beevis D,Bost R, Dring B, Nordo E, Oberman F, Papin JP, Schuffel H, and D Streets. Analysis techniques for human-machine systems design. CSERIAC SOAR 99-01; 1999.

[6] Hardman N, Colombi J, Jacques D, and J Miller. Human systems integration within the DOD architecture framework. Paper presented at IIE Annual Conference and Expo; May 17-21, Vancouver, BC; 2008.

[7] Gawron, VJ. Human performance, workload, and situational awareness measures handbook. 2nd ed. Boca Raton, FL: CRC Press; 2008.

[8] Neville S, Salmon P, Walker G, Baber C, and D Jenkins. Human factors methods: A practical guide for engineering and design. Burlington, VT, USA: Ashgate; 2005.

[9] Wickens, CD. Multiple resources and mental workload. Human Factors, 50(3); 2008. pp 449-455.

[10] Grier, R., C.D., Wickens, David K, et al. The red-line of workload: Theory, research, and design. Paper presented at Human Factors and Ergonomics Society Annual Meeting Proceedings, New York, NY; 2008.

Bibliography

Accot, J., & Zhai, S. (2003). *Refining Fitts' law models for bivariate pointing*. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '03). ACM, New York, NY, USA, 193-200.

Ashdown, M. & Sato, Y. (2005). *Attentive interfaces for multiple monitors.* Institute of Industrial Science, University of Tokyo, Japan.

Card, S. K., Moran, T.P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

DoD. (2012). *Defense acquisition guidebook.* Washington, DC: Headquarters, Department of Defense.

Endsley, M.R. (1988). *Design and evaluation for situation awareness enhancement*. Proceedings of the Human Factors Society 32nd Annual Meeting (Volume 1, pp. 97 – 101). Santa Monica, CA: Human Factors Society.

Feithans, G. L., Rowe, A. J., Davis, J. E., Holland, M., & Berger L. (2008). *Vigilant spirit control station (VSCS)—'The face of COUNTER'.* Proceedings of AIAA Guidance, Navigation and Control Conf. Exhibition, Honululu, HI. AIAA Paper No. 2008-6309.

Fitts, Paul M. (1954): *The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement*. In Journal of Experimental Psychology, 47 pp. 381-391.

Hardman, N. (2009). *An empirical methodology for engineering human systems integration.* (Doctoral Dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH).

Hardman, N., Colombi, J., Jacques, D., & Miller, J. (2009a). *Application of a seeded hybrid genetic algorithm for user interface design*. Proceedings of the IEEE International Conference of Systems, Man, and Cybernetics. San Antonio, TX.

INCOSE. (2007). In Haskins C. (Ed.), *Systems engineering handbook* (ed. 3.1) International Council on Systems Engineering.

Mayhew, D. J. (1992). *Principles and guidelines in software user interface design*. Englewood Cliffs, N.J: Prentice Hall.

Patzek, M., Zimmer, D., Feitshans, G., Draper M., Hughes, T., and Flach, J. (2008). *Multi-uav supervisory control interface technology.* 711th HPW. Dayton, OH.

Rowe, A. J., Liggett, K. K., and Davis, J. E. (2009). *Vigilant spirit control station: A research testbed for multi-UAS supervisory control interfaces*. International Symposium on Aviation Psychology (ISAP).

Schneider, M. & McGrogan, J. (2011). *Architecture based workload analysis of UAS multi-aircraft control: Implications of implementation on MQ-1B predator*. (M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH).

Seibert, M., Stryker, A., Ward, J., & Wellbaum, C. (2010). *System Analysis and Prototyping for Single Operator Management of Multiple Unmanned Aerial Vehicles Operating Beyond Line of Sight.* M.S. Thesis, AFIT/IGSE-06J. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson Air Force Base, Ohio, March 2010.

Stanton, N. A., Salmon, P. M., Walker, G. H., Baber, C., & Jenkins, D. P. (2005). *Human factors methods: A practical guide for engineering and design*. Burlington, VT: Ashgate Publishing Lmtd.

USAF. (2009). *United states air force unmanned aircraft systems flight plan 2009-2047*. Washington DC: Headquarters, United States Air Force.

Wickens, C.D, & Hollands, J. (2000). *Engineering psychology and human performance* (3rd ed.). Upper Saddle River, New Jersey: Pearson Prentice Hall.

Wickens, C.D, Lee, J.D., Liu, Y., & Gordon B. (2004). *An introduction to human factors engineering.* (2nd ed.). Upper Saddle River, New Jersey: Pearson Prentice Hall.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
22-03-2012	Master's Thesis	January 2011 – March 2012

4. TITLE AND SUBTITLE Implementing a Quantitative Analysis Design Tool for Future Generation Interfaces	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Brandon M. Webster, 2Lt, USAF	**5d. PROJECT NUMBER** N/A
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GSE/ENV/12-M08
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Michael Patzek, Ph.D. DSN: 785-7883, Email: michael.patzek@wpafb.af.mil 2210 8th Street, Bldg 146 Air Force Research Laboratory, Wright-Patterson AFB, OH 45433	**10. SPONSOR/MONITOR'S ACRONYM(S)** 711 HPW/RHCI **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The implementation of Multi-Aircraft Control (MAC) for use with Remotely Piloted Aircraft (RPA) has resulted in the need of a platform to evaluate interface design. The Vigilant Spirit Control Station (VSCS), developed by the Air Force Research Laboratory, addresses this need by permitting the rapid prototyping of different interface concepts for future MAC-enabled systems. A human-computer interaction (HCI) Index, originally applied to multi-function displays was applied to the prototype Vigilant Spirit interface. A modified version of the HCI Index was successfully applied to perform a quantitative analysis of the baseline VSCS interface and two modified interface designs. The modified HCI Index incorporates the Hick-Hyman decision time, Fitts' Law time, and the physical actions calculated by the Keystroke-level model. The analysis indicates that the average time for the modified interfaces is statistically less than the average time of the original VSCS interface. These results revealed the effectiveness of the tool and demonstrated in the design of future generation interfaces or modifying existing interfaces.

15. SUBJECT TERMS
Human systems integration (HSI), human-computer interaction (HCI), user interfaces, systems engineering

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					John Colombi, AFIT/ENV
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	57	(937) 255-6565, x 3347 (john.colombi@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18